Claims:

What is claimed is:

1. A method of storing data in a data storage system comprising:

defining a plurality of arrays of equal capacity data storage blocks within said data storage system, each array having a plurality of rows and a plurality of columns such that each column of said plurality of columns corresponds to a separate storage domain of a plurality of storage domains in said data storage system and each row of said plurality of rows corresponds to a plurality of data storage blocks with one data storage block of said plurality of storage blocks from each storage domain of said plurality of storage domains at the same physical address, said plurality of rows having a plurality of contiguous data storage blocks in each said separate storage domain;

defining a plurality of data storage units with at least one data storage unit in each array of said plurality of arrays, each data storage unit of said plurality of data storage units having at least two data storage blocks in different columns of said array of data storage blocks;

creating metadata that includes an array pointer that identifies each array of said plurality of arrays; and

accessing one data storage unit of said plurality of data storage units using said metadata.

2. The method of claim 1 further comprising:

aligning a first row of said array of data storage blocks to a base address in said data storage system that is a multiple of the number of rows in said array of data storage blocks multiplied by said capacity of said data storage blocks.

3. The method of claim 1 where said step of creating metadata further comprises:

defining a second pointer that specifies one data storage unit of a plurality of data storage units in one array or said plurality of arrays.

4. The method of claim 3 where said step of defining a plurality of data storage units further comprises:

defining a plurality of data storage units in each array that each comprise a plurality of data blocks containing user data and that further comprise a plurality of data blocks that contain mirror data if the data storage unit format is RAID-1 and that comprise at least one data block that contains first parity data if the data storage unit format is RAID-5 and that contain second parity data if the data storage unit format is RAID-6. .

5. The method of claim 4 wherein said first parity data is produced by logically exclusive ORing at least two data blocks in the same row of said array.

6. The method of claim 4 wherein said second parity data is produced by logically exclusive ORing at least two data blocks in different rows of said array.

7. The method of claim 4 where user data blocks are stored in a first portion of said array and parity data is stored in a second portion of said array such that at least one row of said array contains no parity data and contains no mirror data.

8. The method of claim 4 further comprising:

defining a sparing table that designates at least one data block in each row of said array as a spare data block.

9. The method of claim 8 further comprising:

defining said sparing table to contain a plurality of sparing versions that each specify a different storage domain as containing said at least one spare data block in each row and such that the sparing version used for an array depends upon the base address of said array.

10. The method of claim 8 further comprising:

defining said sparing table to contain a plurality of sparing versions that each specify a different storage domain as containing said at least one spare data block in each row and such that each sparing version of said plurality of sparing versions is applied to a predetermined number of rows and the sparing version of said plurality of sparing versions used depends upon the base address of said predetermined number of rows.

11. The method of claim 4 further comprising:

organizing the storage domains of said array into a first group and a second group and storing user data associated with a data storage unit in a predetermined number of data blocks in said first group and storing a parity value for said predetermined number of data blocks in said second group if said data storage format is a format that includes parity and storing mirror data in said second group if said data storage format is RAID-1. .

12. The method of claim 4 further comprising:

allocating at least one data storage unit to a logical device.

13. The method of claim 12 further comprising:

creating a metadata entry that specifies said at least one data storage unit and the array of said plurality of arrays in which said at least one data storage unit exists.

14. The method of claim 13 wherein said metadata entry further comprises:

an entry indicating the storage format of said data storage unit.

15. The method of claim 4 further comprising:

creating system metadata that describes the array structure, sparing format, and operating condition of drives in said array.

16. A data storage grid for storing data in a data storage system comprising:

an array of equal capacity data storage blocks defined within said data storage system having a plurality of rows and a plurality of columns such that each column of said plurality of columns corresponds to a separate storage domain of a plurality of storage domains in said data storage system and each row of said plurality of rows corresponds to a plurality of data storage blocks, such that one storage block from each column of said plurality of columns has the same physical address, said plurality of rows having a plurality of contiguous data storage blocks in each domain;

at least one data storage unit defined in said array that corresponds to a predetermined storage format and that specifies the locations in the array of data blocks containing user data and specifies the locations in the array of data blocks containing mirror data if said predetermined storage format is RAID-1 and specifies the location in the array of at least one data block containing row parity if said predetermined storage format is RAID-5 and specifies the location in the array of at least one data block containing vertical parity if said predetermined storage format is RAID-6, said locations in the array of data blocks containing user data organized such that at least one row in said array does not contain mirror data and does not contain row parity and does not contain vertical parity; and

a pointer that identifies said at least one data storage unit.

17. The data storage grid of claim 18 further comprising:

a plurality of data storage blocks designated as spare data storage blocks in said array.

18. A method of converting a data storage grid in a storage system comprising a plurality of data storage grids from a first storage format to a second storage format, said data storage grid having a plurality of data storage blocks configured as a two dimensional array and a predetermined number of data storage units corresponding to said first data storage format forming an ordered set of data storage units defined in said array such that each data storage unit of said plurality of data storage units has a predefined number of user data storage blocks, the method comprising:

identifying said data storage grid and the data storage format of said grid;

accessing metadata for said first data storage format and for said second data storage format that defines the locations and content of data blocks in a grid and that specifies data blocks employed to calculate row parity if the data storage format includes row parity and that specifies the data blocks employed to calculate vertical parity if the data storage format includes vertical parity;

determining the number of data storage units in said first data storage format;

determining the number of data storage units in said second data storage format;

allocating a new grid of the second data storage format if the number of data storage units for said first data storage format exceed the number of data storage formats for said second data storage format;

copying data storage units in excess of the number of data storage units supported by said second format to said new grid if the number of data storage units for said first data storage format exceeds the number of data storage formats for said second data storage format;

copying said data storage units in excess of the number of data storage units supported by said second format to said new grid to create mirror data if the second data storage format is RAID-1 and if the number of data storage units for said first data storage format exceeds the number of data storage formats for said second data storage format;

calculating row parity in said new grid if said second data storage format includes row parity and if the number of data storage units for said first data storage format exceeds the number of data storage formats for said second data storage format;

calculating vertical parity in said new grid if said second data storage format includes vertical parity and if the number of data storage units for said first data storage format exceeds the number of data storage formats for said second data storage format;

designating data storage units in said data storage grid copied to said new grid as free if the number of data storage units for said first data storage format exceeds the number of data storage formats for said second data storage format;

copying data storage units in said data storage grid to create mirror data if said second data storage format is RAID-1;

calculating and writing row parity in said data storage grid as described by said metadata if said second data storage format is RAID-5;

calculating and writing row parity in said data storage grid as described by said metadata if said second data storage format is RAID-6; and

calculating and writing vertical parity in said data storage grid as described by said metadata if said second data storage format is RAID-6.

19. The method of claim 18 further comprising:

determining the logical device to which said data storage grid is allocated;

allocating said new data grid to said logical device; and

updating metadata for said logical device to designate said data storage grid as said second data storage format.

20. A method of sparing a data storage grid in a storage system comprising a plurality of data storage grids in response to a failed storage domain, each data storage grid of said plurality of data storage grids having a plurality of data storage blocks configured as a two dimensional array having a

plurality of columns each corresponding to a separate storage domain of a plurality of storage domains in said storage system and a plurality of rows each corresponding to a data storage block in each said separate domain at the same physical address and having at least one spare data block in each row of said plurality of rows, said method comprising:

identifying said data storage grid in said storage system;

determining the data storage format of said data storage grid;

identifying said failed storage domain;

accessing a sparing table that specifies a domain for said at least one spare data block in each row of said grid;

identifying a non-accessible data block in each row of said grid that corresponds to said failed storage domain;

accessing metadata corresponding to said data storage format that specifies the location and content of each data block in said data storage grid; and

regenerating or copying data corresponding to each said non-accessible data storage block in each row of said grid and storing said data in said at least one spare data block in each row as identified by said sparing table.

21. A data structure that manages access to a storage system having a plurality of data grids each having an array of equal capacity data storage blocks within said data storage system, said array having a plurality of rows and a plurality of columns such that each column of said plurality of columns corresponds to a separate storage domain of a plurality of storage domains in said data storage system and each row of said plurality of rows corresponds to a plurality of data storage blocks, such that one storage block from each column of said plurality of columns has the same physical address, and said plurality of rows having a plurality of contiguous data storage blocks in each domain, said data grids further having a plurality of data storage units each having a predefined group of data storage blocks, said data structure comprising:

a data grid pointer that when multiplied by the storage capacity of said data storage blocks and by the number of rows in said data storage grid and to which an offset, if any, is added, produces the physical address in each storage domain at which said data storage grid begins;

a data grid map that specifies the row and column of each data block of said predefined group of data storage blocks for each data storage unit of said plurality of data storage units; and

a data storage unit pointer selects said row and column of each data block of said predefined group of data blocks for one data storage unit of said plurality of data storage units.

22. The data structure of claim 21 further comprising:

reassignment data indicating if a storage domain had been reassigned.

23. The data structure of claim 21 wherein said data grid map further comprises:

a plurality of map versions that are selected by a portion of the value of said data storage grid pointer.

24. A logical device defined in a data storage system comprising a plurality of data storage units allocated to said logical device, each data storage unit having a predefined group of data storage blocks in a data storage grid, said data storage grid having an array of equal capacity data storage blocks within said data storage system and having a plurality of rows and a plurality of columns such that each column of said plurality of columns corresponds to a separate storage domain of a plurality of storage domains in said data storage system and each row of said plurality of rows corresponds to a plurality of data storage blocks such that one storage block from each column of said plurality of columns has the same physical address, and said plurality of rows having a plurality of contiguous data storage blocks in each domain; said array aligned to a base address in said

data storage system that is multiple of the number of rows in said array multiplied by said capacity of said data storage blocks.

25. A method of accessing data in a data storage system comprising:

defining a plurality of data storage grids in said data storage system, each having an array of equal capacity data storage blocks within said data storage system, said array having a plurality of rows and a plurality of columns such that each column of said plurality of columns corresponds to a separate storage domain of a plurality of storage domains in said data storage system and each row of said plurality of rows corresponds to a plurality of data storage blocks, such that one data storage block from each column of said plurality of columns has the same physical address, and said plurality of rows having a plurality of contiguous data storage blocks in each domain, said data grids further having a plurality of data storage units each having a predefined group of data storage blocks, said array aligned to a base address in said data storage system that is multiple of the number of rows in said array multiplied by said capacity of said data storage blocks;

creating metadata for said plurality of data grids comprising a data grid pointer, a data storage unit pointer, and a data grid map specifying the row and column in said data grid of data blocks comprising said predefined group of data storage blocks for each of said plurality of data storage units;

allocating a plurality of said data storage units to a logical device by specifying a data grid pointer value and a data storage unit pointer value for each data storage unit of said plurality of data storage units allocated;

receiving a request for access to a logical block of said logical device;

determining a domain and a physical address for at least one data storage block by applying said data grid pointer value to said metadata to determine a base address of said data grid, applying said data storage unit pointer value to said data grid map to determine at least one domain to be accessed and to determine a row such that the row number minus one is multiplied by said capacity of said storage blocks and added to said base address to produce a physical address; and

issuing an access command to said at least one domain that includes said physical address.

26. A method of reducing the complexity of data regeneration calculations in a data storage system comprising:

defining a plurality of data storage grids in said data storage system, each having an array of equal capacity data storage blocks within said data storage system, said array having a plurality of rows and a plurality of columns such that each column of said plurality of columns corresponds to a separate storage domain of a plurality of storage domains in said data storage system and each row of said plurality of rows corresponds to a plurality of data storage blocks, such that one data storage block from each column of said plurality of columns has the same physical address, and said plurality of rows having a plurality of contiguous data storage blocks in each domain;

applying a sparing table having a plurality of versions, to said plurality of data storage grids, that each specify at least one data block in each row of each grid as a spare data block to define a plurality of capacity grids, wherein a different version of said plurality of versions is applied to each at least one consecutive grid;

storing data in each capacity grid of said plurality of capacity grids in accordance with a data/parity map;

detecting a storage domain failure; and

excluding at least one data regeneration calculation from at least one capacity grid defined by one version of said plurality of versions of said sparing table wherein said one version designates said failed storage domain as containing said at least one data block specified as a spare data block in said at least one capacity grid.